МГТУ им. Н. Э. Баумана, кафедра ИУ5
курс “Методы машинного обучения”

Лабораторная работа №1

«Создание «истории данных» (Data Storytelling) »

ВЫПОЛНИЛ:
Широков П.Ю.
Группа: ИУ5-21М
Вариант: 15

ПРОВЕРИЛ:
Гапанюк Ю.Е.

Москва 2022

Задание:

• Выбрать набор данных (датасет);

• Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

1. История должна содержать не менее 5 шагов (где 5 – рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию;
2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков;
3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов;
4. Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика;
5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

• Сформировать отчет и разместить его в своем репозитории на github.

Описание набора данных:

Дата - день, месяц, год;

Осадки - кол-во осадков за день;

Максимальная температура - максимальная температура за день;

Минимальная температура - минимальная температура за день;

Ветер - средняя скорость ветра за день;

Погода - оценка дня (солнечный, снежный, дождливый, туманный или с моросящим дождем)

Импортирование необходимых библиотек

```
[3] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/cont
```

```
[19] %%capture
     !wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
     from colab_pdf import colab_pdf
     colab_pdf('ЛР1 - Широков П.Ю. - ИУ5-21М.ipynb')
```

Исследуем основные характеристики датасета

```
[23] data = pd.read_csv("/content/drive/MyDrive/data/seattle-weather.csv")
```

```
data.head()
```

	date	precipitation	temp_max	temp_min	wind	weather
0	2012-01-01	0.0	12.8	5.0	4.7	drizzle
1	2012-01-02	10.9	10.6	2.8	4.5	rain
2	2012-01-03	0.8	11.7	7.2	2.3	rain
3	2012-01-04	20.3	12.2	5.6	4.7	rain
4	2012-01-05	1.3	8.9	2.8	6.1	rain

Веделим новые столбцы, разбив дату

```
[25] data['month'] = data['date'].map(lambda x: int(x[5:7]))
     data['year'] = data['date'].map(lambda x: int(x[:4]))
     data = data.drop(['date'], axis=1)
     data.head()
```

	precipitation	temp_max	temp_min	wind	weather	month	year
0	0.0	12.8	5.0	4.7	drizzle	1	2012
1	10.9	10.6	2.8	4.5	rain	1	2012
2	0.8	11.7	7.2	2.3	rain	1	2012

2	0.8	11.7	7.2	2.3	rain	1	2012
3	20.3	12.2	5.6	4.7	rain	1	2012
4	1.3	8.9	2.8	6.1	rain	1	2012

```
[7] data.shape
```

```
(1461, 7)
```

```
[8] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   precipitation  1461 non-null   float64
 1   temp_max       1461 non-null   float64
 2   temp_min       1461 non-null   float64
 3   wind           1461 non-null   float64
 4   weather        1461 non-null   object 
 5   month          1461 non-null   int64  
 6   year           1461 non-null   int64  
dtypes: float64(4), int64(2), object(1)
memory usage: 80.0+ KB
```

```
data.isnull().sum()
```

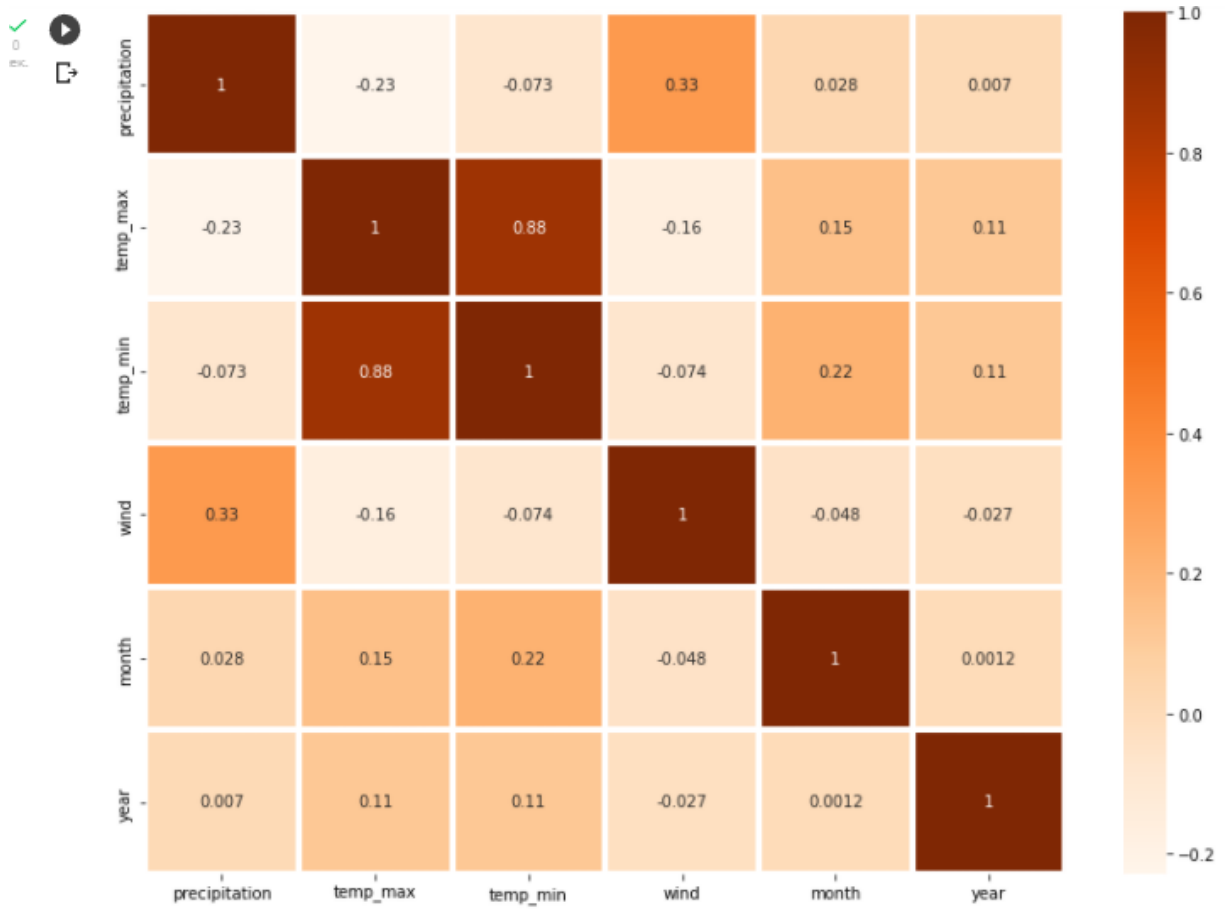
```
precipitation    0
temp_max         0
temp_min         0
wind             0
weather          0
month            0
year             0
dtype: int64
```

```
[10] data['weather'].value_counts()
```

```
rain       641
sun        640
fog        101
drizzle     53
snow        26
Name: weather, dtype: int64
```

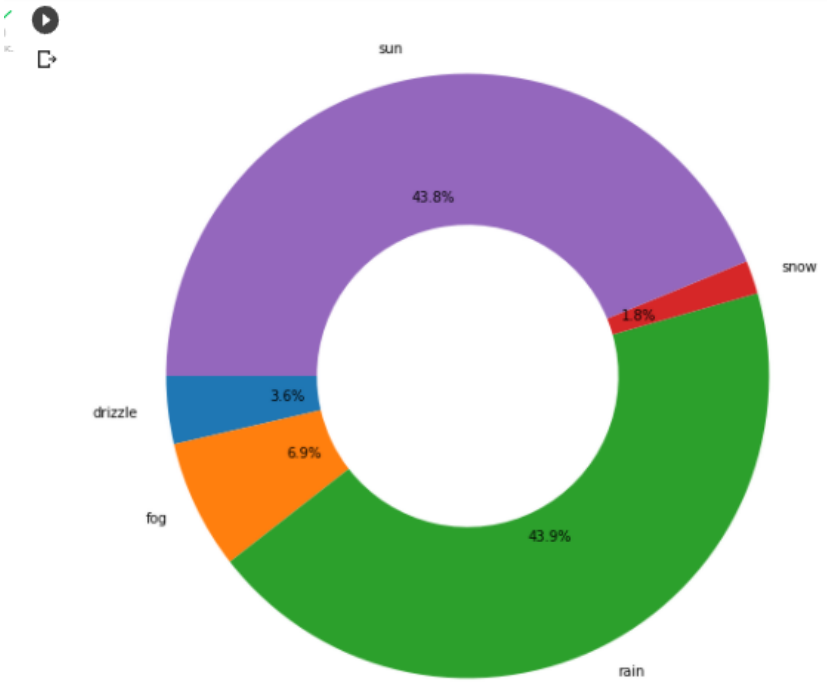
```
[11] plt.figure(figsize=(13,10))
     sns.heatmap(data.corr(), cmap = "Oranges", annot=True, linewidth=3)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe948d50890>
```



Из матрицы корреляции видно, что наиболее сильно коррелируют максимальная и минимальная температуры

```
[18] plt.figure(figsize=(10, 10))
     df = data.groupby(by = 'weather').count()
     labels = df.index
     sizes = df['precipitation']
     plt.pie(sizes, labels=labels, autopct='%1.1f%%', wedgeprops=dict(width=0.5),startangle=180)
     plt.show()
```

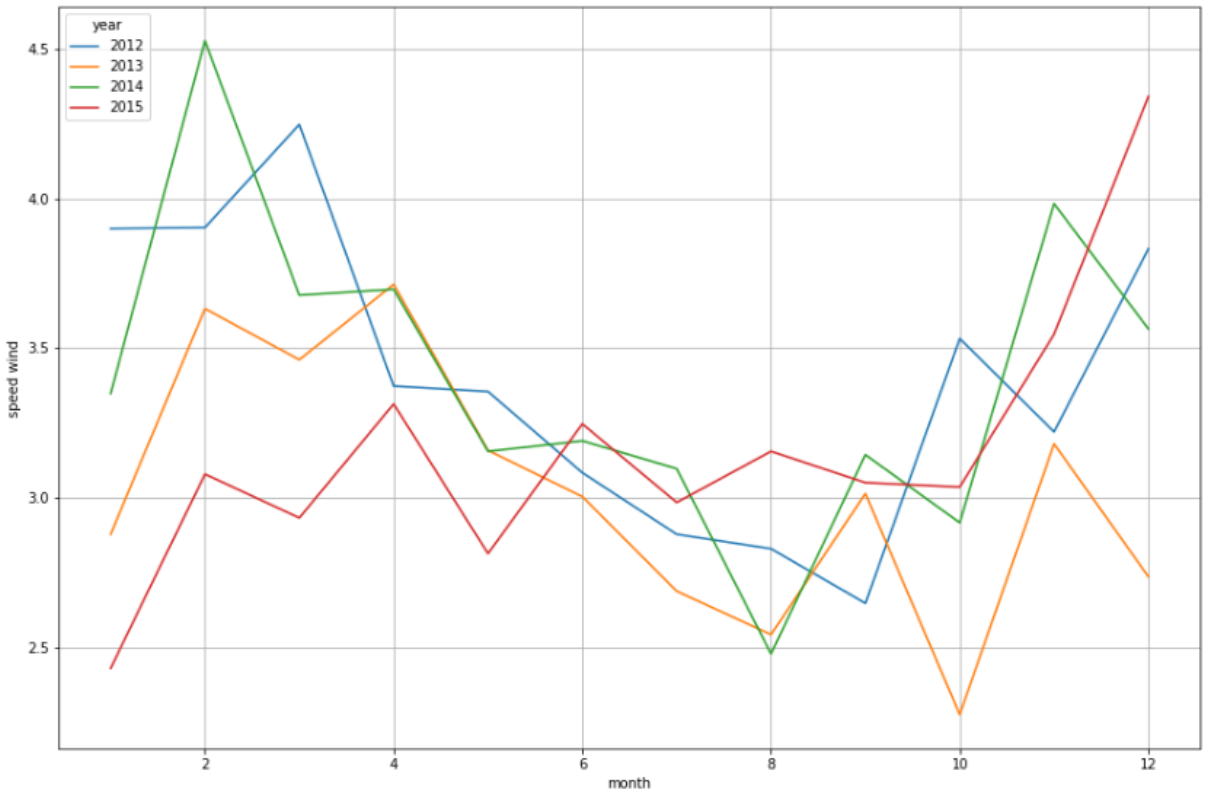


Из круговой диаграммы видно, что в неком городе больше всего дождливых дней, чуть меньше солнечных. А снег там почти не выпадает

```
[26] wind_by_month = data.pivot_table(index=['month'], values='wind', columns='year')
```

```
[26] wind_by_month = data.pivot_table(index=['month'], values='wind', columns='year')
```

```
wind_by_month.plot.line(figsize=(15,10), grid="on");
plt.ylabel("speed wind");
```

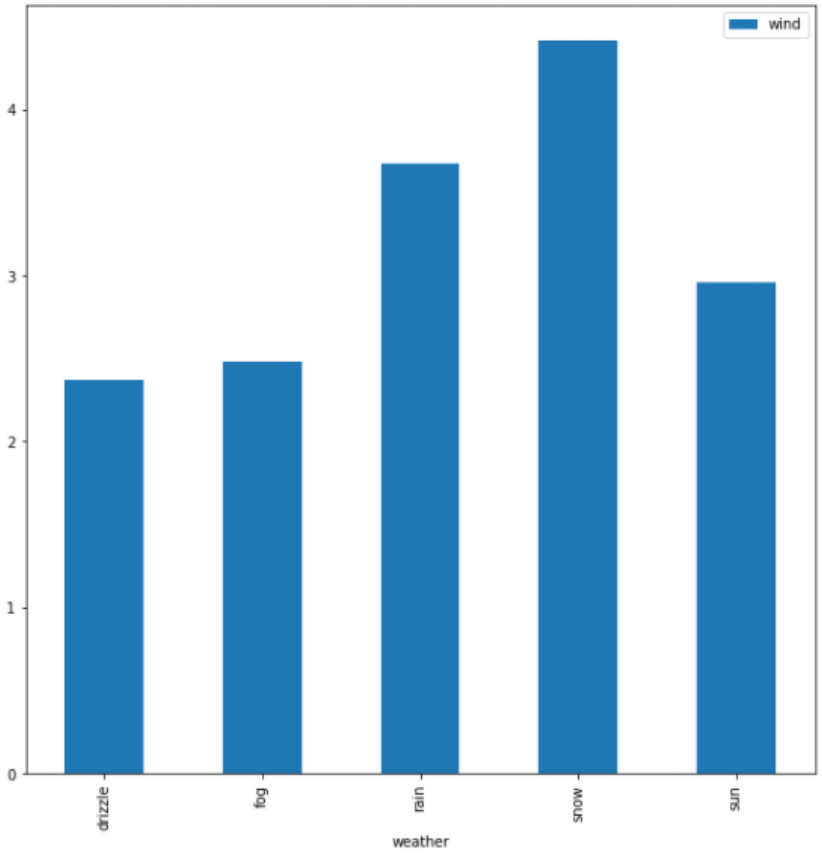


Разбив и сгрупировав среднюю скорость ветра по годам и месяцам можно увидеть ее изменения с 2012 по 2015 года в разне месяцы. Можно заметить, что летом и осенью скорость ветра из года в год мало отличается, а вот весной и зимой погода более непредсказуемая.

```
[ ]
    df1 = data.groupby(by = 'weather').agg({"wind":"mean"})
    df1.plot(kind='bar', figsize=(10,10))
    #sns.plot(x="weather", y="speed wind", data=df1) #, order = data['weather'].value_counts().index)
    #plt.xticks(rotation=90)
```

```
df1 = data.groupby(by = 'weather').agg({"wind":"mean"})
df1.plot(kind='bar', figsize=(10,10))
#sns.plot(x="weather", y="speed wind", data=df1) #, order = data['weather'].value_counts().index)
#plt.xticks(rotation=90)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fef099db450>
```



Из гистограммы видно, что во время выпадения снега наблюдается наибольшая скорость ветра, а во время тумана и мороси наименьшая

```
[14] weather_by_temp = data[['weather','temp_max','temp_min']]
     weather_by_temp['temp_avg'] = (weather_by_temp['temp_max'] + weather_by_temp['temp_min'])/2
     weather_by_temp.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

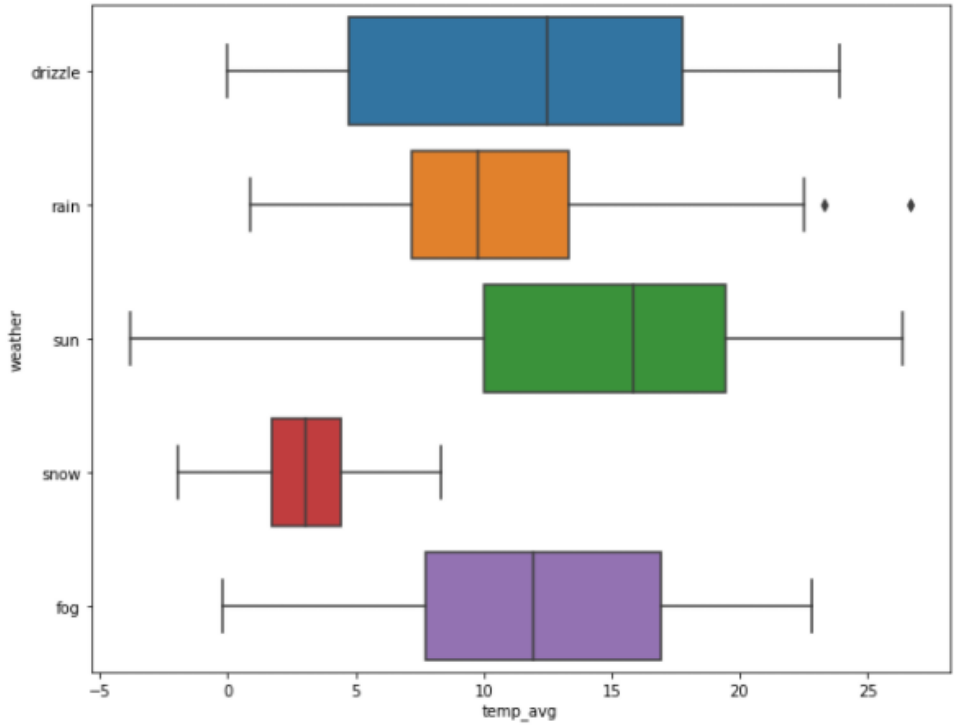
```
[14] weather_by_temp = data[['weather','temp_max','temp_min']]
     weather_by_temp['temp_avg'] = (weather_by_temp['temp_max'] + weather_by_temp['temp_min'])/2
     weather_by_temp.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
```

	weather	temp_max	temp_min	temp_avg
0	drizzle	12.8	5.0	8.90
1	rain	10.6	2.8	6.70
2	rain	11.7	7.2	9.45
3	rain	12.2	5.6	8.90
4	rain	8.9	2.8	5.85

```
plt.figure(figsize=(10, 8))

sns.boxplot(x=weather_by_temp['temp_avg'], y=weather_by_temp['weather'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fef0532f050>
```



Изходя из графика боксплота можно сделать вывод какая средняя дневная температура сопровождает какие дни: солнечные, дождливые, туманные и т.д. Самый большой температурный размах в солнечные дни. Но чаще всего температура колеблется в дни, когда дождь моросит, о чем говорит длинное тело. Самая большая медиана в солнечные дни, самая малая в снежные.

На основании проведенного анализа можно сделать следующий вывод:

- В городе N солнечных и дождливых дней равное кол-во;
- Самый сильный сильный ветер наблюдается во время снега;
- Летом и осенью скорость ветра мало меняется из года в год;
- Если температура выше 14 градусов, то сильный дождь маловероятен.